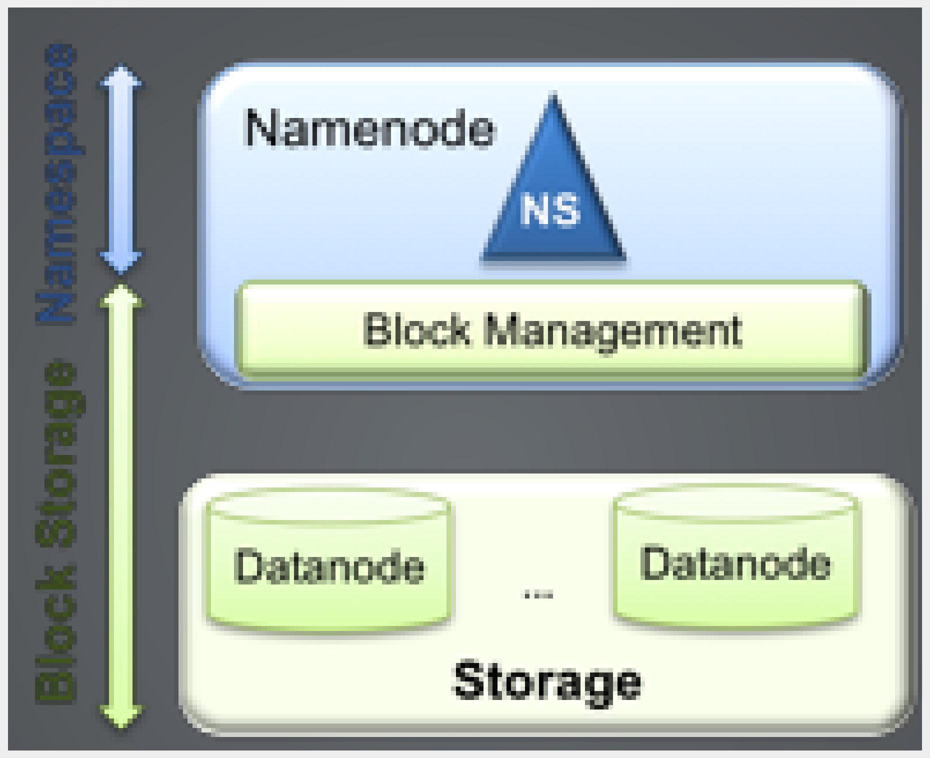
HADOOP

TROUBLESHOOTING

by xiaofei

HDFS块信息存储


Namespace
Block Storage
Namenode
NS
Block Management
Datanode
...
Datanode
Storage

术语

BLOCK

位于NameNode上的Block描述信息

REPLICA

位于DataNode上的副本信息

BLOCK状态

- UnderConstruction
  被create或append的块,block length 和GS未达到最终值
- UnderRecovery
  文件lease到期，由状态UnderConstruction转换到此状态
- Committed
  block的length和GS到达了最终状态.一个未关闭的文件块当NN被新请求一个块时，上一个块由UnderConstruction切换到Committed
- Complete
  complete的block的length和GS是与各个replica的length和GS是完全匹配的。complete只保留finalized replica的位置

BLOCK状态

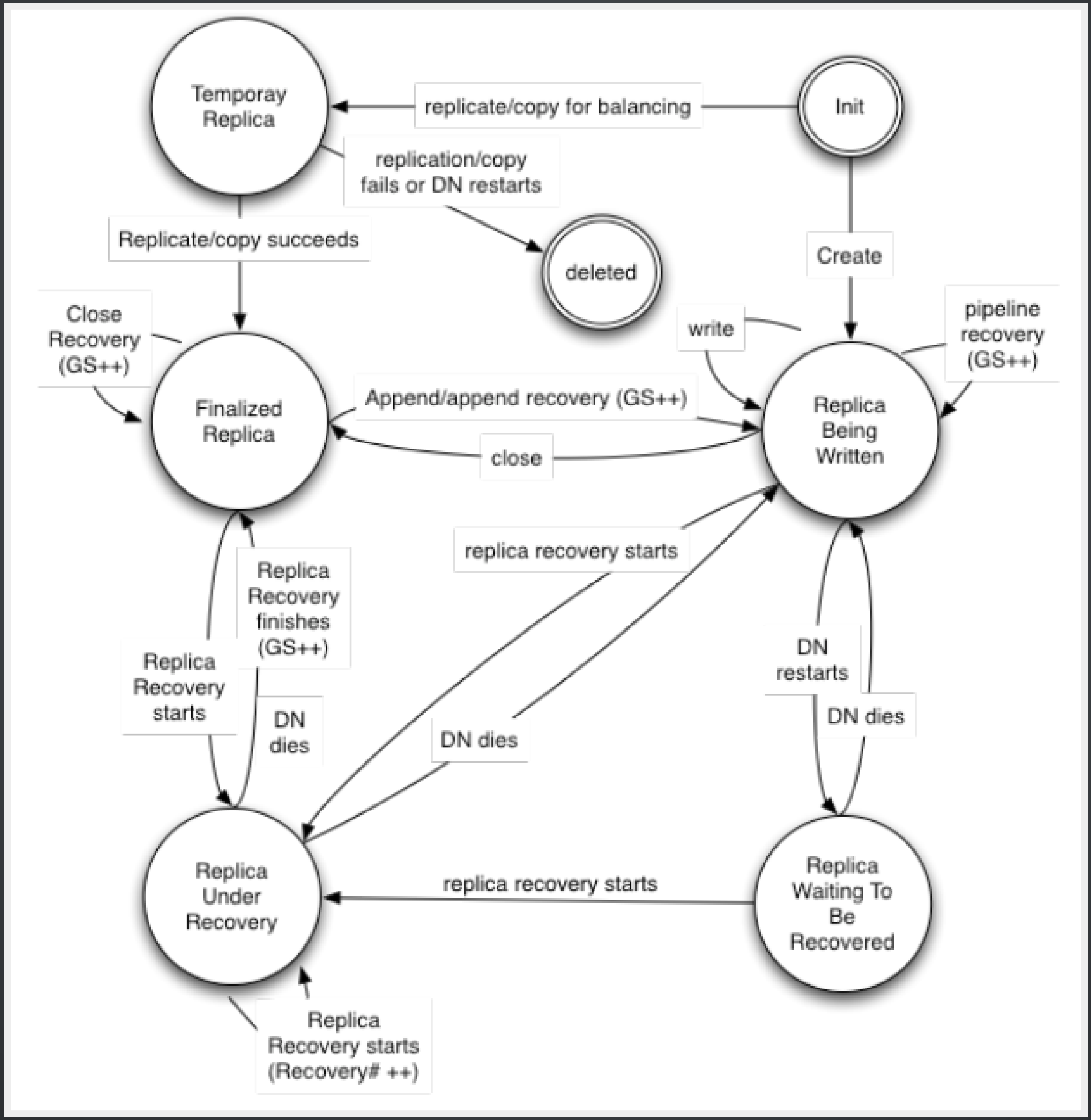
REPLICA状态

- Finalized
  finalized replica的字节已经到达最终状态。新的字节只会在做append操作时才再次写入。但finalized replace的GS不会一陈不变，可能会在做recovery后有变化.
- Rbw(ReplicaBeingWrittento)
  replica create或append后，其位于rbw状态。未关闭文件的最后一个块的状态始终是这个。length和GS未达到最终状态
- Rwr(ReplicaWaitingtobeRecovered)
  当DataNode死掉或重启时，状态为rbw的replica改为rwr。rwr状态的replica不会出现在pipeline中，也不会接收任何其它数据.

REPLICA状态

- rur(ReplicaUnderRecovery)
  当lease过期replica将会将状态改为rur
- Temporary
  temporary状态的replica与replica under construction，但只是由当集群做balance时创建的.它与rwb状态的replica共享很多属性，但数据对用户不可见。在DataNode重启时，位于temporary状态的replica将被删除.

REPLICA状态


Temporay Replica
replicate/copy for balancing
Init
replication/copy fails or DN restarts
Replicate/copy succeeds
deleted
Create
Close Recovery (GS++)
write
pipeline recovery (GS++)
Finalized Replica
Append/append recovery (GS++)
close
Replica Being Written
replica recovery starts
Replica Recovery finishes (GS++)
Replica Recovery starts
DN dies
DN dies
DN restarts
DN dies
Replica Under Recovery
replica recovery starts
Replica Waiting To Be Recovered
Replica Recovery starts (Recovery# ++)

错误处理

- Lease Recovery

LEASE RECOVERY

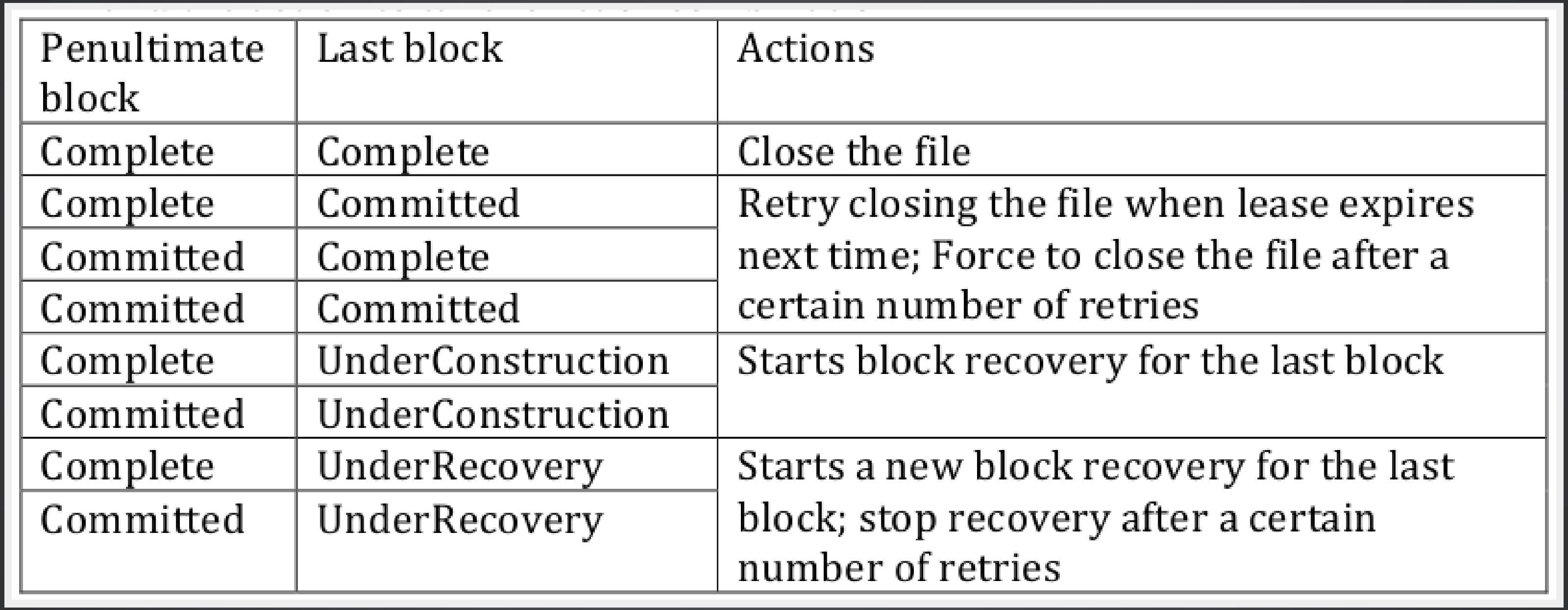
- 并发控制
- 一致性保障

并发控制

NN调用 renewLease(由DFSClient 调用rpc触发)改变文件的leaseholder，同时将每次变更持久化到editlog中。如果client的状态是活动状态的，他的所有与写相关的请求都会请求新的generation stamp。如果没有lease holder像new block,close file操作将被拒绝。这可以防止从client端并发的修改未关闭的文件。

一致性保障

NN会检查文件最后两个block的状态.其它block必须是complete状态。



Penultimate block	Last block	Actions
Complete	Complete	Close the file
Complete	Committed	Retry closing the file when lease expires next time; Force to close the file after a certain number of retries
Committed	Complete	
Committed	Committed	
Complete	UnderConstruction	Starts block recovery for the last block
Committed	UnderConstruction	
Complete	UnderRecovery	Starts a new block recovery for the last block; stop recovery after a certain number of retries
Committed	UnderRecovery	

HADOOP,HBASE错误处理

DFSCLIENT 持续报 COULD NOT COMPLETE FILE .... RETRYING...

```
java.io.IOException: Bad response ERROR for block BP-178649112-10.35.66.1
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer$ResponsePr
2013-11-18 01:15:44,807 WARN org.apache.hadoop.hdfs.DFSClient: Error Reco
6729245 in pipeline 10.35.66.54:50010, 10.35.66.17:50010, 10.35.66.21:500
2013-11-18 01:15:50,129 INFO org.apache.hadoop.hdfs.DFSClient: Could not
2013-11-18 01:15:50,531 INFO org.apache.hadoop.hdfs.DFSClient: Could not
```

DFSCLIENT 持续报 COULD NOT COMPLETE FILE .... RETRYING...

```
//DFSOutputStream
 private void completeFile(ExtendedBlock last) throws IOException {
    long localstart = Time.now();
    boolean fileComplete = false;
    while (!fileComplete) {
      fileComplete = dfsClient.namenode.complete(src, dfsClient.clientNar
      if (!fileComplete) {
        if (!dfsClient.clientRunning ||
              (dfsClient.hdfsTimeout > 0 &&
               localstart + dfsClient.hdfsTimeout < Time.now())) {
            String msg = "Unable to close file because dfsclient " +
                          " was unable to contact the HDFS servers." +
                          " clientRunning " + dfsClient.clientRunning +
                          " hdfsTimeout " + dfsClient.hdfsTimeout;
            DFSClient.LOG.info(msg);
            throw new IOException(msg);
        }
```

DFSCLIENT 持续报 COULD NOT COMPLETE FILE .... RETRYING...

- 严重程度:
  低
- 原因:
  大批量客户端通过DFSClient调用NameNode中的complete完成块的传输调用rpc超过5秒
- 解决办法:

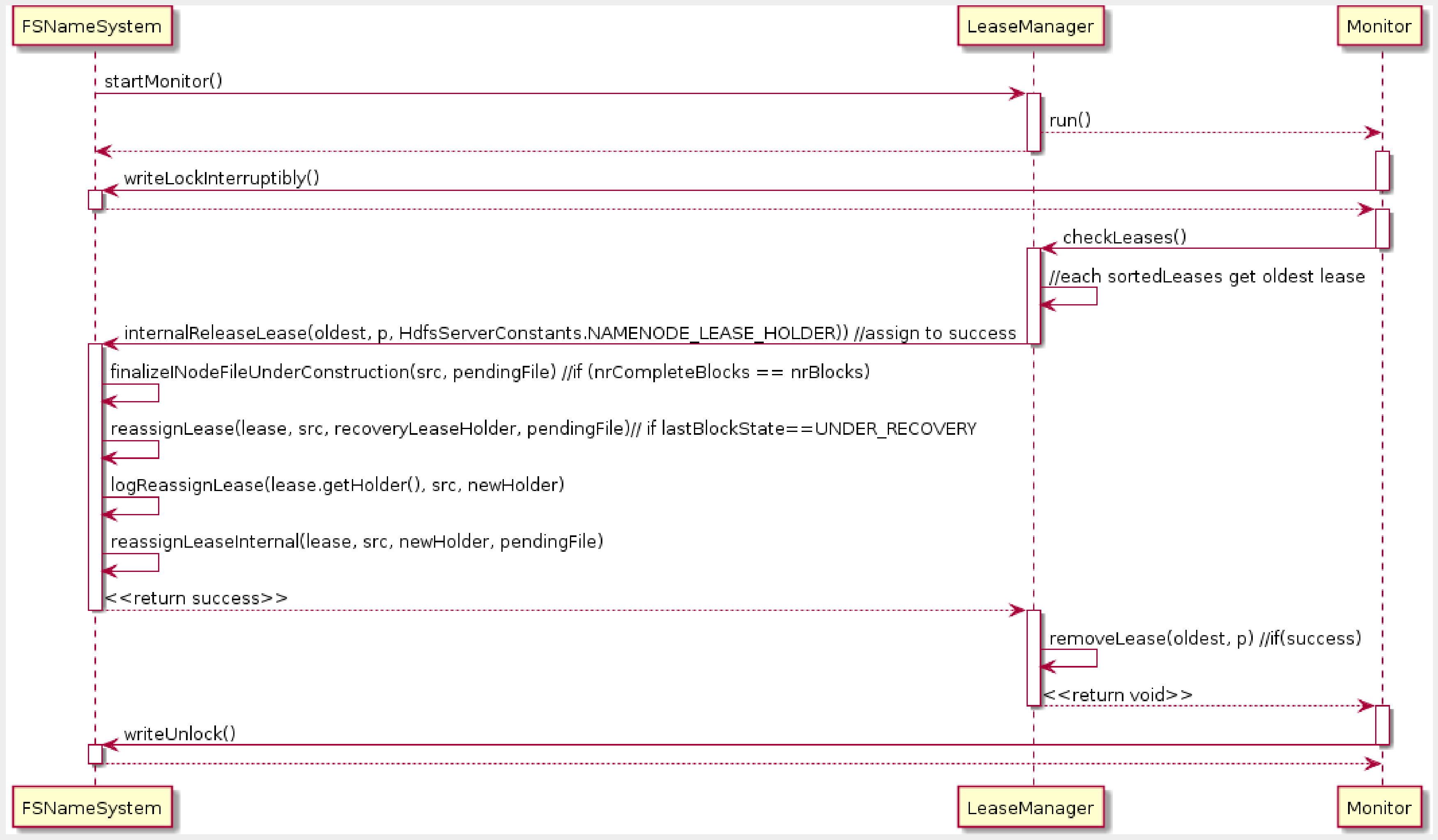
```
<!-- hdfs-site.xml -->
        <property>
                <name>dfs.namenode.handler.count</name>
                <value>512</value>
                <final>true</final>
        </property>
```

NAMENODE持续报BLOCK RECOVER直至NN无响应

```
2013-11-21 18:25:07,534 INFO org.apache.hadoop.hdfs.server.namenode.Lease
2013-11-21 18:25:07,534 INFO org.apache.hadoop.hdfs.server.namenode.Lease
2013-11-21 18:25:07,534 INFO org.apache.hadoop.hdfs.server.namenode.FSNam
```

NAMENODE持续报BLOCK RECOVER直至NN无响应

NAMENODE持续报BLOCK RECOVER直至NN无响应



FSNameSystem

LeaseManager

Monitor

startMonitor()

run()

writeLockInterruptibly()

checkLeases()

//each sortedLeases get oldest lease

internalReleaseLease(oldest, p, HdfsServerConstants.NAMENODE_LEASE_HOLDER)) //assign to success

finalizeINodeFileUnderConstruction(src, pendingFile) //if (nrCompleteBlocks == nrBlocks)

reassignLease(lease, src, recoveryLeaseHolder, pendingFile)// if lastBlockState==UNDER_RECOVERY

logReassignLease(lease.getHolder(), src, newHolder)

reassignLeaseInternal(lease, src, newHolder, pendingFile)

<<return success>>

removeLease(oldest, p) //if(success)

<<return void>>

writeUnlock()

FSNameSystem

LeaseManager

Monitor

NAMENODE持续报BLOCK RECOVER直至NN无响应

```
//LeaseManager Monitor
  class Monitor implements Runnable {
    final String name = getClass().getSimpleName();

    /** Check leases periodically. */
    @Override
    public void run() {
      for(; shouldRunMonitor && fsnamesystem.isRunning(); ) {
        try {
          fsnamesystem.writeLockInterruptibly();
          try {
            if (!fsnamesystem.isInSafeMode()) {
              checkLeases();
            }
          } finally {
            fsnamesystem.writeUnlock();
          }
```

NAMENODE持续报BLOCK RECOVER直至NN无响应

```
//LeaseManager
  private synchronized void checkLeases() {
    assert fsnamesystem.hasWriteLock();
    for(; sortedLeases.size() > 0; ) {
      final Lease oldest = sortedLeases.first();
      if (!oldest.expiredHardLimit()) {
        return;
      }

      LOG.info("Lease " + oldest + " has expired hard limit");

      final List<string> removing = new ArrayList<string>();
      // need to create a copy of the oldest lease paths, becuase
      // internalReleaseLease() removes paths corresponding to empty file
      // i.e. it needs to modify the collection being iterated over
      // causing ConcurrentModificationException
      String[] leasePaths = new String[oldest.getPaths().size()];
```

NAMENODE持续报BLOCK RECOVER直至NN无响应

```
//FSNameSystem
  private void logReassignLease(String leaseHolder, String src,
      String newHolder) {
    writeLock();
    try {
      getEditLog().logReassignLease(leaseHolder, src, newHolder);
    } finally {
      writeUnlock();
    }
    getEditLog().logSync();
  }
```

NAMENODE持续报BLOCK RECOVER直至NN无响应

- 严重程度:
  高
- 原因:
  NN中LeaseManager的Monitor定时检查文件是否硬过期（同时加写锁），如果发现某文件过期则调用FSNameSystem.internalReleaseLease()方法关闭文件，但调用该方法中会触发FSNameSystem.logReassignLease(),同时此方法中也有写锁，造成editlog中的状态不同同步。而interalReleaseLease方法始终返回false,最终功造成死循环。Fix方式见HDFS 4186。

NAMENODE持续报BLOCK RECOVER直至NN无响应

- 解决办法:
- 暂时的避免方案是建议在使用DFSClient时及时关闭操作的文件，不要长时间打开着文件，但不写入任何信息，最终造成NameNode Lease硬过期。
- 长期来看的话需要将当前版本升级到CDH4 4.2.1之后的版本。

HDFS 4186

```
//LeaseManager
class Monitor implements Runnable {
    final String name = getClass().getSimpleName();

    /** Check leases periodically. */
    @Override
    public void run() {
      for(; shouldRunMonitor && fsnamesystem.isRunning(); ) {
        boolean needSync = false;
        try {
          fsnamesystem.writeLockInterruptibly();
          try {
            if (!fsnamesystem.isInSafeMode()) {
              needSync = checkLeases();
            }
          } finally {
            fsnamesystem.writeUnlock();
```

HBASE REGIONSERVER HLOG写入出错，造成REGIONSERVER自动关闭。

```
2013-11-20 01:16:49,124 DEBUG org.apache.hadoop.hbase.regionserver.HRegio
essor.GroupByProtocol
2013-11-20 01:17:02,217 WARN org.apache.hadoop.hdfs.DFSClient: DFSOutputS
460:blk_-4727217747510844304_16938617
java.io.IOException: Bad response ERROR for block BP-178649112-10.35.66.1
0
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer$ResponsePr
2013-11-20 01:17:02,220 WARN org.apache.hadoop.hdfs.DFSClient: Error Reco
6938617 in pipeline 10.35.66.21:50010, 10.35.66.51:50010, 10.35.66.16:500
2013-11-20 01:17:02,315 WARN org.apache.hadoop.hbase.regionserver.wal.HLo
eplicas.  Requesting close of hlog.
2013-11-20 01:17:02,315 DEBUG org.apache.hadoop.hbase.regionserver.LogRol
2013-11-20 01:17:02,330 DEBUG org.apache.hadoop.hbase.regionserver.wal.Se
2013-11-20 01:17:02,330 DEBUG org.apache.hadoop.hbase.regionserver.wal.Se
gs/datanode007.hadoop.bjy.elong.com,60020,1384872281048/datanode007.hadoo
2013-11-20 01:17:07,550 INFO org.apache.hadoop.hdfs.DFSClient: Could not 
datanode007.hadoop.bjy.elong.com%2C60020%2C1384872281048.1384881258958 re
```

HBASE REGIONSERVER HLOG写入出错，造成REGIONSERVER自动关闭。

- 严重程度:
  中
- 问题原因:
  dfs.client.block.write.replace-datanode-on-failure.enable开关未开
  启
- 解决办法:

```
<!-- hdfs-site.xml -->
        <property>
                <name>dfs.client.block.write.replace-datanode-on-failure.enabl
                <value>true</value>
        </property>
```

END